In [34]:
```
import json
import pandas as pd
import numpy as np
from pandas import json_normalize
import matplotlib.pyplot as plt
import seaborn as sns


# Load the JSON data
with open('users.json') as f:
    users_data = [json.loads(line) for line in f]

# Flattening the JSON data
users_flat = json_normalize(users_data)


users_df = pd.DataFrame(users_flat)

# Converting Unix timestamps to datetime objects for further Analysis
users_df['createdDate.$date'] = pd.to_datetime(users_df['createdDate.$da
users_df['lastLogin.$date'] = pd.to_datetime(users_df['lastLogin.$date']
```

In [35]:
```
users_df.head()
```

Out[35]:

	active	role	signUpSource	state	_id.$oid	createdDate.$date	lastLo
0	True	consumer	Email	WI	5ff1e194b6a9d73a3a9f1052	2021-01-03 15:24:04.800	20 15:25:37.85
1	True	consumer	Email	WI	5ff1e194b6a9d73a3a9f1052	2021-01-03 15:24:04.800	20 15:25:37.85
2	True	consumer	Email	WI	5ff1e194b6a9d73a3a9f1052	2021-01-03 15:24:04.800	20 15:25:37.85
3	True	consumer	Email	WI	5ff1e1eacfcf6c399c274ae6	2021-01-03 15:25:30.554	20 15:25:30.59
4	True	consumer	Email	WI	5ff1e194b6a9d73a3a9f1052	2021-01-03 15:24:04.800	20 15:25:37.85

In [36]:
```
# Check for duplicate records
duplicates = users_df.duplicated().sum()
print(f"Number of duplicate records: {duplicates}")
```

```
Number of duplicate records: 283
```

Loading [MathJax]/extensions/Safe.js

In [37]:
```
# Check for missing or null values
missing_values = users_df.isnull().sum()
print(f"\nMissing values:\n{missing_values}")
```

```
Missing values:
active                0
role                  0
signUpSource         48
state                56
_id.$oid              0
createdDate.$date     0
lastLogin.$date      62
dtype: int64
```

In [38]:
```
# Check data types
data_types = users_df.dtypes
print(f"\nData types:\n{data_types}")
```

```
Data types:
active                       bool
role                       object
signUpSource               object
state                      object
_id.$oid                   object
createdDate.$date    datetime64[ns]
lastLogin.$date      datetime64[ns]
dtype: object
```

In [39]:
```
# Identify unique values for categorical columns
unique_roles = users_df['role'].unique()
unique_sign_up_sources = users_df['signUpSource'].unique()
unique_states = users_df['state'].unique()

print(f"\nUnique values in 'role': {unique_roles}")
print(f"Unique values in 'signUpSource': {unique_sign_up_sources}")
print(f"Unique values in 'state': {unique_states}")
```

```
Unique values in 'role': ['consumer' 'fetch-staff']
Unique values in 'signUpSource': ['Email' 'Google' nan]
Unique values in 'state': ['WI' 'KY' 'AL' 'CO' 'IL' nan 'OH' 'SC' 'NH']
```

Loading [MathJax]/extensions/Safe.js

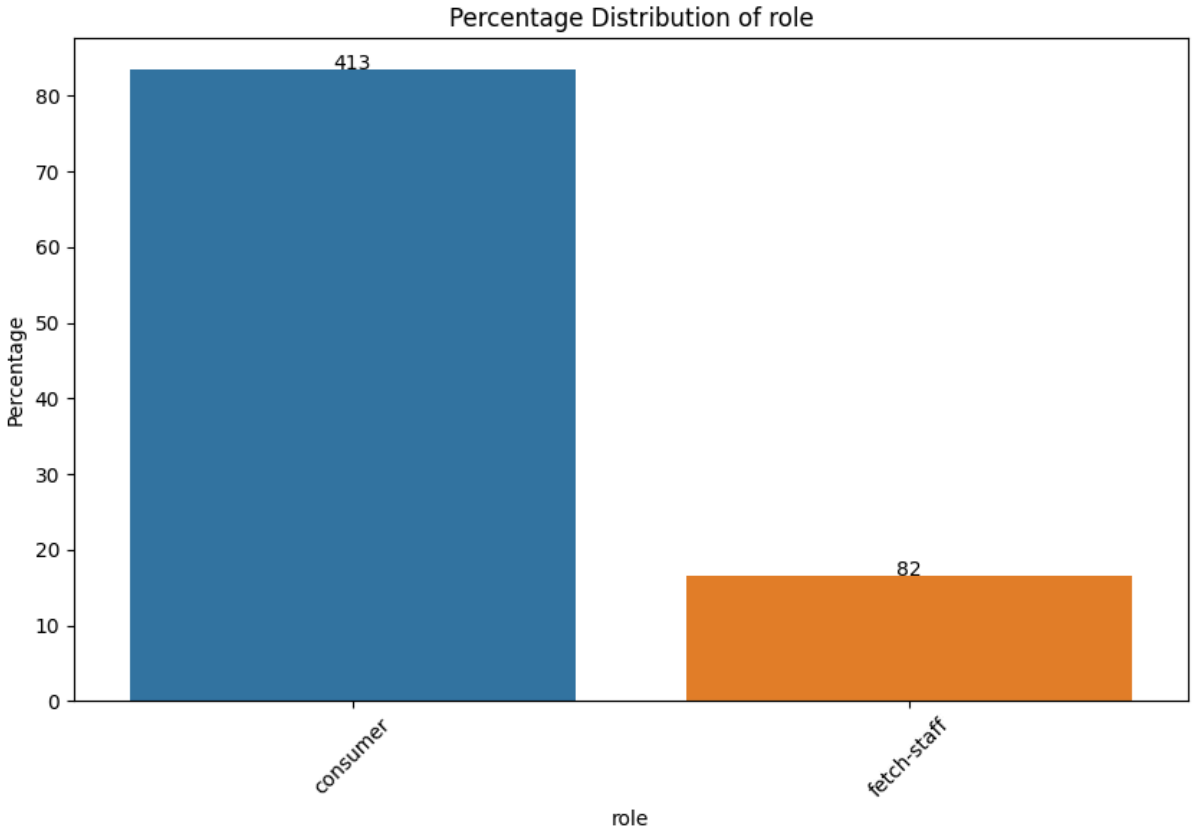
In [40]:
```
# For example, checking if 'createdDate' is before 'lastLogin'
inconsistent_dates = (users_df['createdDate.$date'] > users_df['lastLogi
print(f"\nNumber of records where 'createdDate' is after 'lastLogin': {i
```

```
Number of records where 'createdDate' is after 'lastLogin': 0
```

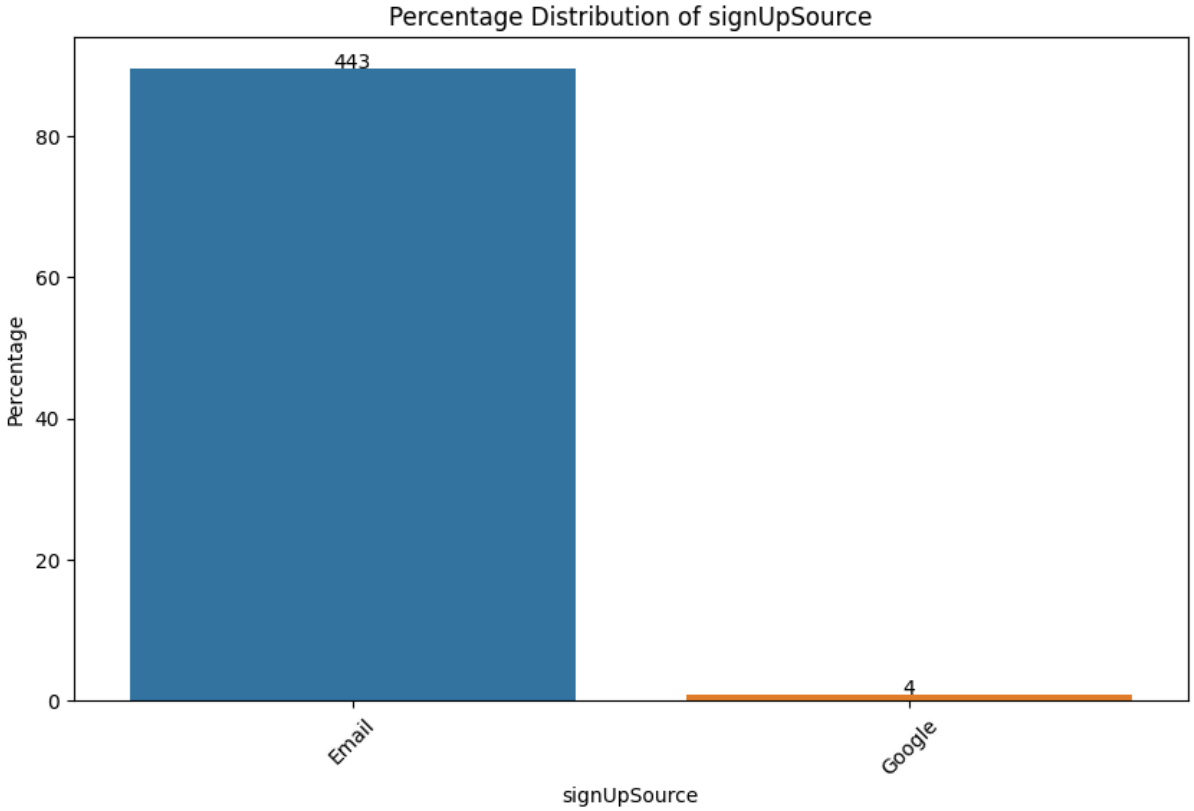
In [ ]:
```

```

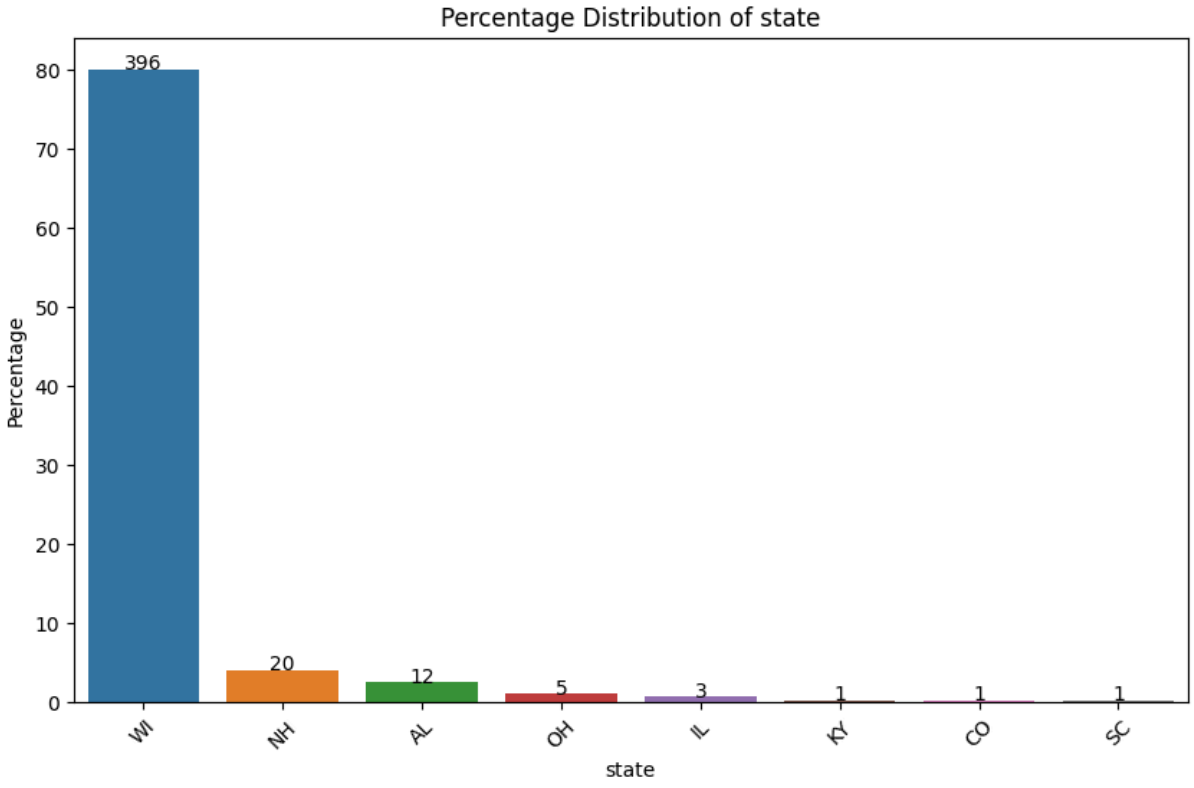
Loading [MathJax]/extensions/Safe.js

In [41]:
```
def plot_percentage_bar_chart(column_name, data_frame):
    total_count = len(data_frame)
    value_counts = data_frame[column_name].value_counts()
    percentages = (value_counts / total_count) * 100

    plt.figure(figsize=(10, 6))
    sns.barplot(x=percentages.index, y=percentages.values)
    plt.title(f'Percentage Distribution of {column_name}')
    plt.xlabel(column_name)
    plt.ylabel('Percentage')
    plt.xticks(rotation=45)

    # Annotate the bars with the actual numbers
    for index, value in enumerate(value_counts):
        plt.text(index, percentages.values[index], f'{value}', ha='cente

    plt.show()

# Plot percentage bar charts for 'role', 'signUpSource', and 'state'
plot_percentage_bar_chart('role', users_df)
plot_percentage_bar_chart('signUpSource', users_df)
plot_percentage_bar_chart('state', users_df)
```



Loading [MathJax]/extensions/Safe.js





In [ ]:

The Data quality issues I found in the users.json-

1. Some missing Values in the signUpSource and LastLogin Columns.
2. There were duplicate records in the dataset, leading to redundancy.

Loading [MathJax]/extensions/Safe.js

3. Data Imbalance in the Categorical Categories which can be due to data collection biases or anomalies.
4.

In [ ]:

Loading [MathJax]/extensions/Safe.js